

Olli Lääkkö

KONFIGURAATTORIPROJEKTIN LÄPIVIENTI

3D-mallin konfigurointi Excelillä

Tekniikan ja luonnontieteiden tiedekunta
Kandidaatintyö
Joulukuu 2019

TIIVISTELMÄ

Olli Lääkkö: Konfiguraattoriprojektin läpivienti

Kandidaatintyö

Tampereen yliopisto

Konetekniikka

Joulukuu 2019

Konfiguraattorit mahdollistavat tuotteiden yksilöllisen suunnittelun kunkin asiakkaan tarpeisiin ilman, että suunnitteluorganisaation edustajien tarvitsee osallistua erikseen myyntiprosessiin. Konfiguraattorin avulla yritys voi sekä parantaa tuotteen laatua ja resurssien tehokkaampaa käyttöä, että vähentää henkilökunnan työkuormaa ja poistaa motivaatiota heikentäviä toisteisia työtehtäviä. Konfiguraattoreita voidaan rakentaa erilaisten päättelymallien varaan. Käyttökohde vaikuttaa konfiguraattorin päättelymallin valintaan.

Konfiguraattorin rakentamisen perustaksi vaaditaan yrityksen tuottaman tuotevalikoiman ja tuotteiden välisten yhteyksien tarkkaa tuntemusta. Tätä tuotetiedon hallintaa varten on olemassa useita erilaisia sovelluksia, joita ei käsitellä tässä työssä sen tarkemmin. Konfiguraattoria varten luotavien tietomallien muokkaaminen ja kehittäminen vaatii yrityksen prosessien havainnollistamista, mikä itsessään voi jo auttaa yrityksen sisäisten prosessien kehityksessä.

Konfiguraattoriprojekteihin liittyy useita yrityksen ulkoisia ja sisäisiä haasteita, jotka tekevät konfiguraattorin maksimaalisen hyödyn saavuttamisen hankalaksi. Yritysten sisäiset haasteet liittyvät pääasiassa erilaisiin johtamisen haasteisiin ja yrityksen sisäisen tiedonkulun heikkouksiin, kun taas ulkoiset haasteet ovat pääasiassa asiakastarpeen ja oikean asiakassegmentin ymmärtämisen puutteita.

ABSTRACT

Olli Lääkkö: Configurator project implementation

Bachelor's Thesis

Tampere University

Mechanical engineering

December 2019

Configurators make customer-specific projecting possible without engineering in the selling process. The company can increase the quality of products and use their resources more efficient by using the configurator. Same time the company decreases the planning organization's workload and removes repeated works that decrease motivation. Configurators can be based on a couple of different deduction models. The right model depends on the application.

Successful configurator project is based on good knowledge about company product range and relationships between products in different assemblies. There is a lot of different product data management software in the market, but I don't handle these in this thesis. Good knowledge representations must be created before building the configurator itself. This visualization process itself might help the organization to make its internal processes better.

There are a lot of internal and external challenges in carrying out the project. These challenges are making full profit achieving very difficult. The reasons for the internal challenges are mainly in leadership and in-house communication. External challenges include mainly problems to understand customer demands or choose the right customer segment.

SISÄLLYSLUETTELO

1. JOHDANTO	1
1.1 Työn tausta	1
1.2 Työn tavoitteet	2
1.3 Työn rakenne	2
2. KONFIGUROIINNIN PERUSTEET	4
2.1 Tuotetiedon hallinta	4
2.2 Konfiguroinnin päättelymallit	5
2.2.1 Sääntöihin perustuva päättely	5
2.2.2 Tapauksiin perustuva päättely	5
2.2.3 Malliin perustuva päättely	6
3. KONFIGURAATTORIT	8
3.1 Konfiguraattorin luominen	8
3.1.1 Kehitysstrategiat	10
3.1.2 Konfiguraattoriprojektien haasteet	13
3.2 Konfiguraattorin tuoma lisäarvo	14
4. SOVELTAVA OSUUS	17
4.1 Soveltavan työn rajaaminen	17
4.2 Konfiguraattorin luominen	17
5. YHTEENVETO	21
LÄHTEET	23

1. JOHDANTO

1.1 Työn tausta

Kuluttajien kysyntä laajempaa tuotevalikoimaa kohtaan on kasvanut (Bonev et al., 2015). Tuotevalikoiman lisääminen ja kustomointi kasvattavat yrityksen varastoinnin tarvetta ja sitoo pääomia. Tuotevalikoiman laajentaminen massakustomoinnin avulla saa pidettyä kustannukset matalina, koska tuotteiden muokkaaminen toteutetaan kevyemmin metodein, eikä valmistusmäärien tarvitse olla niin suuria. Massakustomoinnin myötä myös asiakastyytyväisyys ja markkinaosuus kasvaa, ymmärrys asiakkaista paranee ja yritys tuottaa paremmin (Selladurai, 2004). Tuottava teollisuus on siis osin siirtynyt aikakauteen, jolloin tuotteisiin halutaan massatuotannon mahdollistama edullinen hinta ja hyvä laatu, mutta myös mahdollisimman hyvin asiakkaan haluamaan tarpeeseen kohdistetut ominaisuudet. Uniikkien tuotteiden tapauksessa asiakastarpeisiin vastaaminen voidaan määritellä alusta alkaen hyvin tarkasti, mutta tuotteen hinta kasvaa ja jokaisella kerralla valmistetaan ikään kuin uusi prototyyppi, kun toistojen kautta tulevaa laadun parantamista ei ole.

Konfiguraattorit ovat sovelluksia, jotka auttavat tuotevalikoiman kustomoinnissa ja luovat selkeän viitekehityksen mahdollisista muutoksista (Trentin et al., 2012). Konfiguraattorin avulla aiemmin mainitut kustomoinnit ja tuotteen kaikki mahdolliset ominaisuudet voidaan viestiä asiakkaalle selkeästi ja inhimilliset virheet vähenevät sekä tuotekehityksessä että myyntiprosessissa. Lähtökohtaisesti on toki aina parempi, jos tuotteet voidaan standardoida tiettyihin vaihtoehtoihin, jolloin tuottavan yrityksen ei tarvitse hallita niin useita tuotevariantteja. Mikäli standardointi ei ole mahdollista, on tuotteen konfiguroiminen järkevää tuotteen suunnitteluun liittyvien päällekkäisyyksien poistamiseksi. Mikäli tuoteperheen elinkaari on kohtalaisen lyhyt, on hyvä, että konfiguraattorin luominen ja uusiminen olisi mahdollisimman kevyt prosessi.

1.2 Työn tavoitteet

Konfiguraattorien tuomat hyödyt on helppo ymmärtää, mutta silti läheskään kaikki konfiguraattoriprojektit eivät päädy valmiiksi tuotteiksi (Blazek & Pils, 2017). Tässä työssä on tarkoitus käsitellä kirjallisuustutkimuksen keinoin haasteita ja vaatimuksia, jotka täytyy ratkaista ennen valmiin konfiguraattorin käyttöönottoa. Konfiguraattorin luominen vaatii yritykseltä hyvän tuotetiedon hallinnan, jonka päälle konfiguraattoria voidaan lähteä suunnittelemaan. Vaikka yritys ei olisi luomassa konfiguraattoria, epäselvä tuotetiedon hallinta heikentää yrityksen tehokkuutta, sillä samoja tuotteita voidaan suunnitella useita kertoja, kun ei tiedetä aiempien vastaavien olemassaolosta.

Työn tavoitteena on löytää vastaus tutkimuskysymyksiin: Mitä vaaditaan onnistuneeseen konfiguraattoriprojektiin? Mitä onnistuneella konfiguraattoriprojektilla voidaan saavuttaa? Työ tarkastelee konfiguraattoreita kompleksisissa tuotekokonaisuuksissa ja keskisuurissa sekä suurissa organisaatioissa. Pienet organisaatiot ja tuotteet tarjoavat järjestelemätöntä joustoa lyhyiden tietoketjujen ja pienten tietomäärien vuoksi, jolloin ihmisten tiivis vuorovaikutus paikkaa järjestelmässä olevia puutteita.

1.3 Työn rakenne

Työ koostuu viidestä luvusta, joista tämä johdanto on ensimmäinen. Johdannon jälkeen käydään läpi konfiguroinnin perusteita. Tuotetiedon hallinta on keskeinen rakenneosia toimivan konfiguraattorin luomisessa. Kun yrityksellä on ymmärrys käsittelemistään komponenteista ja niiden välisistä yhteyksistä, he myös voivat luoda selkeästi hallittavat rakenteet erilaisten kokoonpanojen luomiseksi. Tässä luvussa käydään läpi myös erilaisia konfiguroinnin päättelymalleja, joiden mukaan konfiguraattorit luovat asiakkaan vaatimusten mukaisia kokoonpanoja.

Kolmas luku keskittyy konfiguraattoreihin. Millaisilla strategioilla konfiguraattoreita kehitetään ja mitkä ovat minkäkin strategian vahvuudet ja heikkoudet. Samalla käydään läpi myös yleisimpiä haasteita ja epäonnistumisen mahdollisuuksia konfiguraattoriprojekteissa. Luvun lopussa selvitetään myös konfiguraattorien tuomia hyötyjä ja niiden toteutumiseen vaikuttavia haasteita yrityksen sisäisen johtamisen ja markkinatarpeen tunnistamisen näkökulmasta.

Neljäs luku on soveltava osio, jossa luodaan konfiguraattori yksittäiselle tuotteelle. Soveltavan osion tarkoituksena on tuoda teoriaa hieman käytäntöön. Toki aiemmissa luvuissa esitelty teoria keskittyy enemmän yrityksiin, joissa tuotekokonaisuudet ja käsiteltävät tietomassat ovat suuria. Soveltava osio syventää kuitenkin ymmärrystä

konfiguroinnin perusasioista. Viidennessä luvussa on yhteenveto, jossa tiivistetään lyhyesti työn tuomat oivallukset koko prosessista.

2. KONFIGUROINNIN PERUSTEET

2.1 Tuotetiedon hallinta

Tuotetiedon hallinta on tuotteen kehitykseen liittyvän tarkan tuotetiedon hallintaa ja tämän tarjoamista sitä tarvitseville työkaluille ja järjestelmille oikea-aikaisesti tuotteen koko elinkaaren ajan (Crnkovic et al., 2003 s.19). Useat suunnittelijat tekevät nykyaikaista suunnittelua ilman, että suunnittelijat sijaitsisivat samalla konttorilla tai edes samassa valtiossa. Tämä hajautettu rinnakkainen suunnittelumalli yhdistettynä konfiguroitavaan tuotevalikoimaan on pakottanut yritykset kehittämään ja miettimään omaa tuotetiedon hallintaansa. PDM (Product data management) eli tuotetiedon hallinta on tärkeä osa suunnittelutyön sujuvuutta, ja siksi yrityksillä onkin PDM-järjestelmänsä ylläpitämässä tuotteisiin liittyvää tietokantaa. (Peltonen et al., 2002 s.9) PDM:n ja PDM-järjestelmien ero on siinä, että tuotetiedon hallinta on yleinen nimi säännöille, periaatteille ja metodeille, joiden pohjalta toimintoja suoritetaan eri järjestelmissä. PDM-järjestelmä on kokoelma työkaluja tuotetietojen ja prosessien hallintaan. (Crnkovic et al., 2003 s.20)

Tuotetiedon hallinnalla tarkoitetaan yleisesti kaikkea tuotteeseen liittyvään informaatiota tuotteen rakenteellisista tiedoista myynnillisiin tietoihin. Tuotetiedon hallinnan voidaan ajatella jakautuvan viiteen järjestelmien tukemaan kyvykkyyteen, joita ovat tietovarasto nimikkeille ja dokumenteille, työnkulun ja prosessien hallinta, tuoterakenteiden hallinta, osien hallinta ja toimintojen hallinta tuotannon rytmittämiseen (Liu & Xu, 2001). PDM-järjestelmät lähestyvät asiaa enemmän tuotesuunnittelun näkökulmasta, jolloin niihin talletetaan yleensä vain tuotteiden teknisiin ominaisuuksiin liittyvää informaatiota. Kokonaisvaltaisempia toiminnanohjausjärjestelmiä kutsutaan usein ERP- tai ERM-järjestelmiksi (Enterprise Resource Planning / Management) ja niihin on yhdistetty myös tuotteen myyntiin ja markkinointiin sisältyvää dataa. (Peltonen et al., 2002 s.11)

Nimikkeet ja niiden hallinta muodostavat perustan tuotetiedon hallinnalle. Nimikkeitä ovat yleisellä tasolla kaikki itsenäiset osat ja komponentit. Käytössä olevat nimikkeet ovat PDM-järjestelmän perusosia ja määrittelevät sen, mitä tietoja järjestelmällä voidaan käsitellä. (Peltonen et al., 2002 s.15; Crnkovic et al., 2003 s.22) Nimikkeiden hallinta tuottaa tiedon kaikista yrityksen käyttämistä komponenteista, mutta niiden väliset yhteydet löydetään erilaisista tuoterakenteista. Tuotteen tärkeimpänä rakenteena voidaan pitää suunnittelurakennetta, sillä se sisältää kaikki osat lopullisessa kokoonpanossa mukaan lukien mahdolliset kuljetuksen tai varastoinnin vaatimat väliaikaiset osat. (Martio, 2015)

s.116) Tuotetiedon hallinta on tärkeä pohjatieto konfiguraattoria luotaessa, jotta ymmärretään käytettävissä olevat mahdollisuudet erilaisiin tuotekokoonpanoihin.

2.2 Konfiguroinnin päättelymallit

Konfigurointia voidaan toteuttaa usealla eri päättelymallilla. Riippuu tuotteen rakenteesta ja konfiguraattorin käyttäjäryhmästä minkälaisen päättelyn varaan konfiguraattori tulisi rakentaa. Konfigurointiteoria jakaa konfiguraattorit kolmeen pääryhmään päättelymallien mukaan (Martio, 2015 s.42).

2.2.1 Sääntöihin perustuva päättely

Sääntöihin perustuva päättely perustuu ohjelmointikieliin, jotka tukevat jotain tekoälyä. Tekoälyä hyödyntävien konfigurointijärjestelmien ensimmäisiä versioita käytettiin jo 1970-luvulla, jolloin konfiguraattorin toiminta perustui toteutuskelpoisuuden mukaan järjesteltyihin sääntölausekkeisiin. Sääntöihin perustuvan päättelyn suurin haaste on sääntöjen suuri määrä ja jatkuva muuttuminen, mikä tekee ohjelmointiin perustuvien järjestelmien ylläpidosta erittäin haastavaa. (Martio, 2015 s.42) Järjestelmän käytettävyyden kannalta on olennaista, että järjestelmän ylläpitoon käytettävät resurssit on huomattavasti pienempi kuin järjestelmän antama hyöty.

Parametrien laskemista ja komponenttien valintaa voidaan myös toteuttaa CAD-ohjelmien yhteyteen saatavilla makropohjaisilla ohjelmointikielillä. Näissäkin haasteena on tiedon selkeä esittäminen, sillä ohjelmamassa kasvaa varsin suureksi ja tuotteeseen liittyvät säännöt on sisällytetty tähän koodiin. (Martio, 2015 s.43) Järjestelmän ylläpito ohjelmistoissa vaatii ylläpitäjältä hyvän ymmärryksen käytetystä kielestä ja tuotteesta. Optimaalisessa tilanteessa ylläpito saadaan tehtyä mahdollisimman vähäisellä tietotaidolla ja sen myötä automatisoitua.

2.2.2 Tapauksiin perustuva päättely

Tapauksiin perustuva päättely vaatii toimiakseen suuren määrän ennakkotapauksia. Kun konfiguraattoriin annetaan lähtötiedot uudesta konfiguraatiosta, järjestelmä käy läpi kaikki aiemmin tehdyt konfiguraatiot lähtötietoineen ja luo uuden tuloksen näiden pohjalta. Konfiguraation oikeellisuudesta ei ole uusien konfigurointitulosten tapauksessa täyttä varmuutta, koska järjestelmälle ei ole annettu sääntöjä, joiden mukaan päättelyä tulisi suorittaa. (Martio, 2015 s.43)

Tapauksiin perustuva päättelymalli ei siis toimi elinkaaren alkupäässä oleville tuotteille, koska se vaatii toimiakseen suuren ennakkotapausten joukon. Siitä syystä tämän päättelyn käyttäminen tuotteiden ja palveluiden konfiguroinnissa ei ole toimivaa. Sen sijaan lääketieteellisessä diagnostiikassa tästä päättelymallista voisi olla hyötyä, jossa tapauksina käytettäisiin oireisiin perustuvia diagnooseja. (Martio, 2015 s.43) Tapauksiin perustuvan päättelyn tietoisuus on siis yhtä suuri kuin sille syötetty tietomassa. Tuotekehityksen tapauksessa puhuttaessa suurin osa konfiguraattorien käyttökohteista sijoittuu elinkaaren alkupäähän ja uuden luomiseen, eli tapauksiin perustuvan päättelyn hyödyntäminen ei ole niin yleistä.

2.2.3 Malliin perustuva päättely

Malliin perustuvan päättelyn tavoitteena on ollut poistaa sääntöihin perustuvan päättelyn monimutkaisuuden aiheuttamat ylläpito-ongelmat. Mallit sisältävät elementtirakenteen ja elementtien väliset riippuvuudet, joiden pohjalle rakentuu järjestelmien keskeisin runko. Malliin perustuvia järjestelmätyyppejä ovat logiikkapohjainen päättely, rajoitepohjainen päättely ja resurssipohjainen päättely, joita voidaan käyttää myös yhdessä. (Martio, 2015 s.43) Erilaisten päättelymuotojen yhdistäminen tehostaa malliin perustuvan päättelyn käyttöä ja mahdollistaa käytön useammissa tilanteissa.

Logiikkapohjainen päättely hyödyntää tavallisten aritmeettisten ja loogisten lausekkeiden lisäksi formaalisia logiikkakieliä, jotka on kehitetty tietämyksen esittämiseen. Järjestelmän rakenne perustuu olioihin, joissa yksittäinen olio sisältämiensä attribuuttien kanssa sisältyy olioryhmään, jolla on joukko yhteisiä attribuutteja. Tämän lisäksi olioiden välillä on roolijako, joka kertoo olioiden välisistä yhteyksistä. Konfigurointiohjelma tulkitsee näitä järjestelmän sisältämiä sääntöjä sekä attribuutteja ja luo konfiguraation, joka noudattaa rooleja ja olioiden ominaisuuksia annetut lähtötiedot toteuttaen. (Martio, 2015 s.44) Konfiguraation muodostuminen kuulostaa hyvin samalta, kuin sääntöihin perustuvassa päättelyssä, mutta erona on mahdollisiin yhdistelmiin liittyvän tiedon sijoittuminen olioihin, jolloin ohjelman rakenne on helpommin hallittavissa.

Rajoitepohjaisessa päättelyssä rajoitteiden avulla määritellään olioiden mahdollisista yhteyksistä. Järjestelmä etsii annettujen lähtötietojen perusteella kaikki mahdolliset konfiguraatiot, jotka rajoitteiden pohjalta ovat mahdollisia. Parametrien antojärjestys voi olla mielivaltainen, mikä helpottaa käyttöä ja on myyntikonfiguraattoreiden tapauksessa välttämätöntä, mikäli konfiguraattoria käyttää henkilö, joka ei tunne tuotetta etukäteen. Tämä johtuu siitä, että valintapolkujen luominen vapaasti valittaville parametreille tuottaa

suuren määrän mahdollisia polkuja jo kohtalaisen pienellä parametrimäärällä. Kymmenen vapaasti valittavaa parametriä tuottaa jo yli 350 000 valintapolkua. Rajoitepohjaisuus vapauttaa tuotemallin laatijan siis luomasta näitä kaikkia valintapolkuja, vaan ohjelma tarkastelee luotujen rajoitteiden toteutumista valintajärjestyksestä riippumatta. Rajoitepohjaisessa päättelyssä lopputuloksen oikeellisuus on helppo arvioida ehtojen toteutumisen kautta. (Martio, 2015 s.44) Rajoitepohjaisella päättelyllä toteutetussa konfiguraattorissa käyttäjä voi lisätä lähtötietoja, jotka karsivat mahdollisten varianttien määrää vähemmäksi.

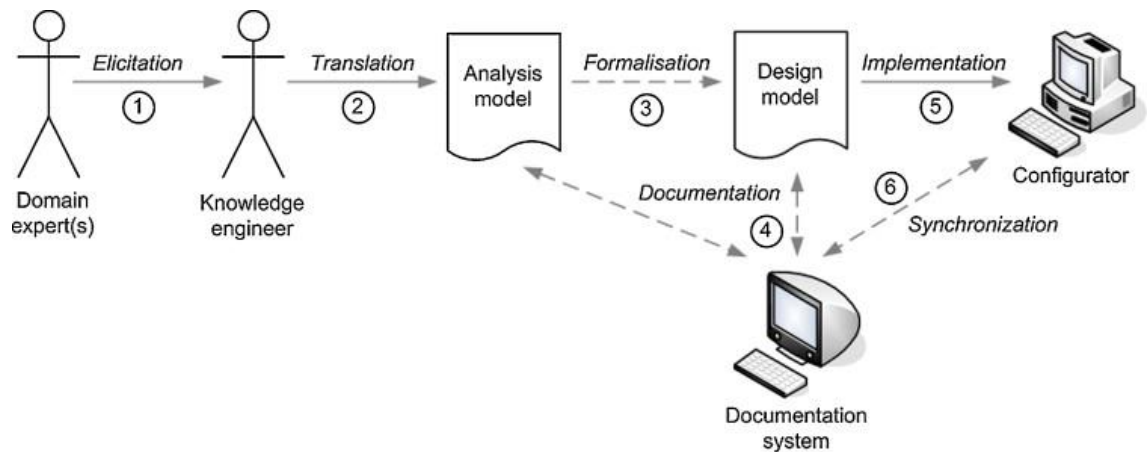
Resurssipohjaisessa päättelyssä jokainen olio sisältää tiedon sen tuottamasta tai kuluttamasta resurssista. Tätä päättelymallia voidaan käyttää lisänä konfiguroitaessa tuotteita, joissa esimerkiksi on komponentteja, jotka tuottavat ja kuluttavat sähköä, paineilmaa, voimaa tai muuta energiaa. Resurssipohjainen päättely varmistaa, että tuotanto riittää kattamaan kulutuksen halutussa kokonaisuudessa. (Martio, 2015 s.45) Resurssipohjaisen päättelyn käyttö on hyvä lisä yhdisteltäessä näitä päättelymalleja, jolloin mekaanisen yhteensopivuuden lisäksi saadaan tieto esimerkiksi generaattorin tehon riittävydestä.

3. KONFIGURAATTORIT

Konfiguraattorit ovat sovelluksia, joilla voidaan helpottaa osien tai laitteiden räätälöintiä tilanteissa, joissa yrityksen valikoimassa on paljon ennalta määrättyjä mahdollisuuksia koostaa valmis tuote (Crnkovic et al., 2003 s.42; Trentin et al., 2012). Ne ovat asiantuntijajärjestelmiä, jotka tukevat käyttäjää tuotteiden räätälöinnissä (Martio, 2015 s.42; Hvam et al., 2019). Konfiguraattoreita käytetään yrityksen tuotevalikoiman tiedottamisessa asiakkaalle, asiakkaan haluaman tuotevariantin kuvauksen ja tuotetietojen tarkistamiseen, jotta yritys pystyy sen asiakkaalle toimittamaan, reaaliaikaisen tiedon toimittamiseen tuotevariantin hinnasta, kustannuksista, teknisistä ominaisuuksista ja toimitusajasta, tarjousten tekemiseen sekä asiakkaan haluamia ominaisuuksia vastaavan tuotevariantin määrittelemiseen (Trentin et al., 2012). Lyhentyneet läpimenoajat, muunneltujen tuotteiden parempi laatu, parempi tietämys tuotevalikoimasta, tuotteiden optimointiin käytetyn ajan ja rutiinitehtävien vähentäminen, toimintavarmuus ja työntekijöiden perehdyttämiseen tarvittavien resurssien pieneneminen ovat esimerkkejä hyödyistä, joita konfiguraattoreilla voidaan saavuttaa (Haug et al., 2012).

3.1 Konfiguraattorin luominen

Haug et al. (2012) esittävät julkaisussaan konfiguraattorin luomisen jakautuvan kuuteen vaiheeseen, jotka näkyvät kuvassa 1. Kuvassa katkoviiva analyysimallin ja suunnittelumallin välillä kuvaa, ettei näiden kahden välistä eroa ole tarkasti määritetty. Pisteviivat dokumentointijärjestelmään kuvaavat, että dokumentointijärjestelmää ei sisällytetä välttämättä jokaiseen vaiheeseen.



Kuva 1 Konfiguraattorin kehityksen vaiheet (Haug Anders et al. 2012 s.475)

Konfiguraattorin toteuttamisen ensimmäinen vaihe on sovellukseen tulevan tiedon hankinta. Tiedon hankinta ajatellaan nykyisin olemassa olevan tiedon kasaamisen sijaan jatkuvasti kehittyvänä mallina, jota muokataan uuden tietämyksen kertyessä. (Haug et al., 2012) Tämä tieto voidaan ajatella myös reaaliaikaisen tietona, jonka esitystapa ei välttämättä ole kovin muodollinen (Hvam et al., 2019).

Toisessa vaiheessa muodostetaan saadun tiedon perusteella analyysimalli, jonka avulla voidaan havainnollistaa konfiguraattorin luomiseen tarvittavia tuotetietoja. Haug et al. (2012) ja Hvam et al. (2019) käyttävät tästä termiä ilmiömalli, joka sisältää tuotteen rakenteen ja toimintojen lisäksi tuotteen elinkaaren. Tärkeintä tässä vaiheessa on luoda riittävän havainnollinen malli, joka ilmentää käsiteltyjä prosesseja. Tämän pohjalta konfiguraattoria työstävä asiantuntijat luovat mallinnusympäristön rakenteisiin sopivan mallinsa.

Kolmas vaihe on tarpeellinen, mikäli analyysimallin rakenne eroaa paljon toteutettavan konfiguraattorin mallinnusympäristön rakenteista. Samalla voidaan lisätä mahdolliset integraatiot muihin järjestelmiin, mikäli tarpeellista. (Haug et al., 2012; Hvam et al., 2019)

Neljäntenä vaiheena on dokumentointi, jolla koostettu suunnittelumalli saadaan myös prosessin ulkopuolisten ihmisten käyttöön ja arvioitavaksi (Haug et al., 2012). Dokumentoinnin heikkolaatuisuus on riski konfiguraattorin onnistumiselle (Hvam et al., 2019).

Viidennessä vaiheessa suunnittelumallin pohjalta luodaan konfiguraattori valitulla alustalla joko suoraan mallista tai viimeisten muutosten kanssa (Haug et al., 2012; Hvam et al., 2019).

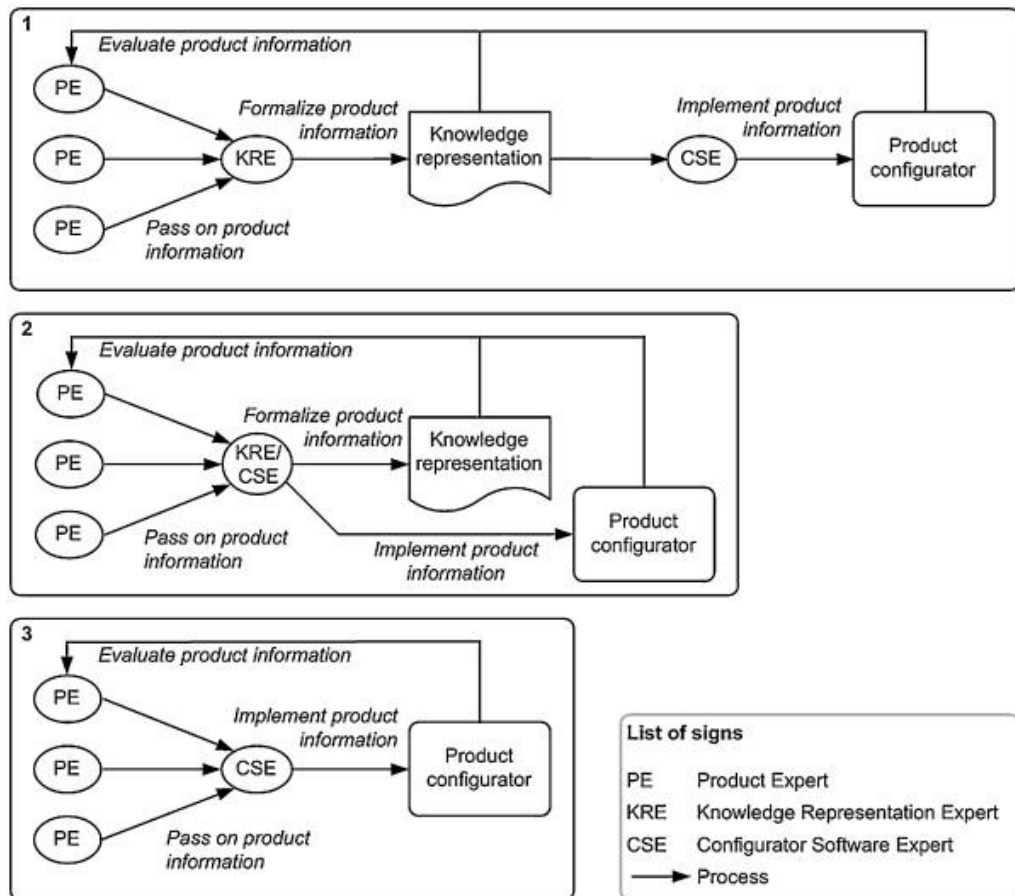
Kuudentena vaiheena on synkronointi, jonka tarkoituksena on päivittää dokumentaatio ajan tasalle aina kun konfiguraattorissa tapahtuu muutoksia (Haug et al.,

2012). Dokumentaation ylläpidon puutteet ovat yksi konfiguraattoriprojektien epäonnistumisia aiheuttavista tekijöistä, koska yleiskuva jää heikoksi ja muutosten tekeminen jälkikäteen on vaikeaa (Hvam et al., 2019).

Konfiguroitavien tuotteiden kompleksisuuden ja hallittavan nimikemäärän kasvaessa on UML-pohjaisten mallinnustekniikoiden käyttö tehokkaampaa verrattuna ei-UML - pohjaisiin (Hvam et al., 2019). UML on oliomallinnustekniikkaan pohjautuva graafinen mallinnuskieli, joka sisältää rakenteen, käyttäytymisen ja vuorovaikutuksen kuvaamiseen soveltuvia kaavioita (Luukkainen & Laine, 2010).

3.1.1 Kehitysstrategiat

Konfiguraattorien luomista varten on olemassa erilaisia kehitysstrategioita, jotka vaikuttavat konfiguraattoriprojektin kustannuksiin, valmistumisaikaan ja valmiin konfiguraattorin laatuun. Kuvassa 2 on kuvattu kolmen yleisimmin käytetyn strategian toiminta. Strategiat eroavat toisistaan tiedonkulun ja projektiin liitettyjen henkilöiden osalta. (Haug et al., 2012) Eri strategioiden heikkouksista ja vahvuuksista kerron lisää tämän luvun loppupuolella.



Kuva 2 Konfiguraattoriprojektin kehityksen kolme päästrategiaa (Haug Anders et al. 2012 s.477)

Strategian 1 mukaisessa toteutuksessa jokaiselle aihealueelleen on nimetty oma asiantuntijansa. Prosessin tuotoksena olevat tuotetiedon mallit käyvät tuoteasiantuntijoiden arvioitavana ennen niiden käyttöönottoa lopullisessa konfiguraattorissa.(Haug et al., 2012) Tässä strategiassa prosessiin käytetty organisaatio on suurin ja projektin eteneminen muihin strategioihin verraten hidasta. Toisaalta näkökulmat ja tietomallit tulee tarkasteltua huolella ennen konfiguraattorin luomista.

Strategiassa 2 tuotetietoja keräävät ja tuotetietomalleja luovat samat ihmiset, jotka tekevät lopullista konfiguraattoria. Tässäkin toteutuksessa tuoteasiantuntijoilla on mahdollista arvioitua luotuja malleja ennen konfiguraattorin luomista ja sen aikana. (Haug et al., 2012) Tämän strategian haasteeksi näkisin sellaisten ihmisten löytämisen, jotka sekä osaavat luoda selkeitä visuaalisia malleja prosesseista että hallitsevat konfiguraattorihjelmistojen käytön.

Strategia 3 on suoraviivaisin ja keveimmällä asiantuntijaorganisaatiolla toteutettu. Siinä ei tuoteta erillisiä käsitteellisiä malleja, vaan oikeanlaiset riippuvuudet ja yhteensopivuudet on todettava konfiguraattoria käyttämällä. Näistä päästrategioista käytetään myös useita muunnelmia ja erityisesti jo valmiiksi saadun konfiguraattorin ylläpidossa käytettävät strategiat poikkeavat hieman näistä päästrategioista. (Haug et al., 2012) Kolmannen strategian haasteeksi näkisin erillisen tietomallin puuttumisen. Tuotetietomallin kokeilu ja kehitys konfiguraattorin kautta ja konfiguraattoria muokkaamalla ei ole yhtä helppoa kuin erillisen suunnittelumallin muokkaaminen. Tämä strategia vaatii tuoteasiantuntijoilta ja sovelluskehittäjältä erinomaista kommunikointia ja yhteistä ymmärrystä asiasta.

Anders et al. ovat arvioineet tekemänsä tutkimuksen perusteella edellä esitettyjä päästrategioita seitsemällä eri kriteerillä, jotka on esitelty taulukossa 1. Taulukossa on vertailtu kolmen strategian eroja toisiinsa nähden.

Taulukko 1 Päästrategioiden keskinäinen vertailu. Mukailten (Haug Anders, Hvam Lars et al. 2012 s.478)

	Suhteelliset edut (+) ja haitat (-)	S1	S2	S3
1.	Kerätyn tuotetiedon arviointi ennen käyttöönottoa	+	+	-
2.	Käyttöönotetun tuoteinformaation muuttamisen helppous	+	+	-
3.	Konfiguraattorihjelmiston vaihtamisen helppous	+	+	-
4.	Kommunikoinnin helpottaminen tuoteasiantuntijoiden ja konfiguraattoriasiantuntijoiden välillä	+	-	-
5.	Dokumentointiin käytettyjen resurssien minimointi	-	-	+

6.	Tiedonsiirron minimointi (vähemmän väärinkäsityksiä ja virheitä)	-	+	+
7.	Prosessin sujuva toteutus	-	+	+

Taulukon 1 mukaan kerätyn tiedon arviointi ennen käyttöönottoa on strategian 1 ja 2 vahvuus ja strategian 3 heikkous. Suurimmalta osaltaan tämä johtuu siitä, että kolmannessa strategiassa kerätyn datan pohjalta ei luoda erillisiä käsitteellisiä malleja. Mallit helpottavat tuoteasiantuntijoiden osalta tiedon oikeellisuuden varmistamista ja konfiguraattorin kasaajan on huomattavasti helpompi ymmärtää tuotekokonaisuuteen liittyvät riippuvuudet ja rajoitteet rakennetun mallin avulla. Kolmannen strategian mukaan toteutettu konfiguraattori voi myös vähentää omaa uskottavuuttaan, mikäli testausvaiheessa käyttäjät joutuvat monesti toteamaan puutteita ja epäkohtia konfiguraattorissa. Tämä voi heikentää valmiin konfiguraattorin käyttöastetta. (Haug et al., 2012) Konfiguraattorin uskottavuuden heikkeneminen on yksi konfiguraattoriprojekteihin liittyvistä sisäisistä haasteista, joista enemmän seuraavassa luvussa.

Käsitteellisten mallien ja dokumentoinnin puute vaikuttaa myös taulukossa toisena olevaan arviointikriteeriin, joka on käyttöönotetun tuoteinformaation muuttamisen helppous. Usein konfiguraattoria muokkaavat henkilöt eivät ole itse tuotteen asiantuntijoita ja päinvastoin tuotteen asiantuntijat eivät välttämättä ymmärrä itse konfiguraattorin toimintaa. Kun yhteistä kuvausta tai dokumentointia ei ole, on uusien tietojen lisääminen ja vanhojen muokkaaminen työläämpää kuin mallien kanssa. (Haug et al., 2012) Mallit toimivat siis selkeänä ja mielellään mahdollisimman visuaalisena taustamateriaalina, joihin konfiguraattorin kehittäjät voivat palata.

Kolmas arviointiperuste, konfiguraattorihjelmiston vaihtamisen helppous, on myös strategioiden 1 ja 2 vahvuus. Useimmissa konfiguraattorihjelmistoissa on omat kielensä tietokantojen kirjoittamiseen, eli yksi ajantasainen ulkoinen tietomalli ja -rakenne helpottaa ohjelmiston vaihtoa ja vähentää prosessissa tulevia virheitä. (Haug et al., 2012) Konfiguraattorihjelmiston vaihtaminen voi olla ajankohtaista yrityksen kasvaessa, markkinatilanteen muuttuessa tai teknologioiden kehittyessä ohi aiemmista.

Kommunikoinnin helpottaminen tuoteasiantuntijoiden ja konfiguraattoriasiantuntijoiden välillä lukeutuu ainoastaan strategian 1 eduksi. Strategian 1 vahvuus tällä osa-alueella on tehtävien selkeä jakaminen eri alojen asiantuntijoille sekä se, että mukana on erillinen henkilö, joka kerää ja koostaa tiedon selkeään muotoon ennen kuin se päättyy konfiguraattorihjelmistoasiantuntijoille. Tämä tuoteasiantuntijoiden ja ohjelmistoasiantuntijoiden väliin tuleva henkilö mahdollistaa sen, että ohjelmiston luojat saavat keskittyä täysin ohjelman luontiin ja muokkaukseen palautteen perusteella. (Haug et al., 2012) Tässä strategiassa on myös helpointa löytää

riittävän pätevät henkilöt jokaiseen rooliin, kun asiantuntijaroolit ovat niin selkeästi rajattuja.

Viides osa-alue on selkeästi strategian 3 vahvuus. Tätä strategiaa valitessa tulee vahvasti harkita, onko dokumentoinnin pois jättämisestä saadut säästöt aidosti isommat kuin dokumentoinnin tuomat hyödyt. (Haug et al., 2012) Joissain lyhyen elinkaaren tai muuten pienen tuotteen kohdalla voi hyvinkin olla.

Kaksi viimeistä osa-aluetta, tiedonsiirron minimointi sekä prosessin sujuva toteutus, ovat strategioiden 2 ja 3 vahvuuksia. Tämä perustuu lyhyempään tietoketjuun, eli prosessissa on mukana vähemmän henkilöitä, jolloin tiedon vaihtamisen tarve on vähäisempi. (Haug et al., 2012)

Sopivan strategian valinta on riippuvainen toteuttavan organisaation koosta sekä tuotteen monimutkaisuudesta. Strategioiden 1 ja 2 teknisessä toteutuksessa erona on dokumentoinnin laajuus. Kun strategiassa 2 samat ihmiset sekä koostavat tiedon, että toteuttavat konfiguraattorin, ei heidän tarvitse dokumentoida kaikkea yhtä suurella laajuudella kuin strategiassa 1. Mikäli löydetään yksittäisiä henkilöitä, joilla on riittävä taito tuotetiedoista, tiedon dokumentoinnista ja esittämisestä, on strategia 1 on varmin valinta. Strategia 3 on kevein, nopein ja edullisin toteuttaa, mutta samalla riskialtis, mikäli tuote on monimutkainen tai siihen on mahdollisesti tulossa useita muutoksia, jolloin dokumentoinnin puutteesta johtuvat ongelmat muuttuvat suuremmiksi kuin edellä mainitut hyödyt. Strategia 2 on näiden välillä oleva toteutus, jossa tiedonkeruun ja konfiguraattorin toteuttamisen yhdistelmällä voidaan saavuttaa nopeampaa toteutusta ja keveämpää dokumentointia ilman, että muokattavuus kuitenkaan kärsisi merkittävästi. Eri strategioiden yhdisteleminen voi olla hyödyllistä projektin edetessä. Esimerkiksi strategialla 1 aloitettu projekti voi jatkua ensimmäisen todellisen mallin testauksen jälkeen strategialla 2 tai 3, jolloin viimeiset korjaukset saadaan toteutettua nopeammin. (Haug et al., 2012)

3.1.2 Konfiguraattoriprojektien haasteet

Konfiguraattorit tuottavat onnistuessaan monia hyötyjä käyttäjilleen, mutta silti monet konfiguraattoriprojektit keskeytyvät tai lopetetaan jo ennen varsinaista toteutusta. Näissä tapauksissa tilanne voi olla se, että organisaatio tai markkina ei ole vielä valmis käyttämään konfiguraattoria ja se toteutetaan täysin vastaavan oloisessa organisaatiossa hieman myöhemmin. (Haug et al., 2012) Mikäli konfiguraattorin käyttäjänä on asiakas suoraan, voi haasteeksi muodostua tuotesuunnitteluparametreihin pohjautuva konfiguraation määrittely, minkä käyttämiseen asiakkaan asiantuntemus ei välttämättä riitä (Wang et al., 2015).

Konfiguraattoriprojektin onnistumiseen liittyvät haasteet voivat tulla sekä konfiguraattoria käyttävän organisaation sisältä, että yrityksen ja markkinan välisestä suhteesta. Sisäisesti konfiguraattoriprojektin onnistumisen kannalta on merkittävää, että projekti etenee riittävän nopeasti ja alkaa tuottamaan hyötyä yritykselle mahdollisimman pian, jotta projektiin laitettavat rahalliset ja työajalliset investoinnit näyttäytyvät järkevinä yrityksen johtotasolle (Haug et al., 2012). Projektin valmiiksi saamisen kannalta on tärkeää, että projektille on nimetty vastuullinen johtaja (Blazek & Pilsl, 2017). Kun konfiguraattori on saatu valmiiksi, on yrityksen sisäisenä haasteena vielä saada konfiguraattori käyttöön kattavasti yrityksen sisällä. Edellytys sisäisen käyttöönoton osalta on, että konfiguraattori kattaa riittävän laajasti valmistetun tuotevalikoiman, jotta sillä voidaan luoda tarpeeksi tarkkoja ja laajoja kokonaisuuksia. Konfiguraattorin ylläpitoon on myös varattava resursseja, jotta kaikki tuotevalikoimassa tapahtuvat muutokset tulee päivitettyä konfiguraattoriin ja siten koko organisaation tietoisuuteen. (Haug et al., 2012) Sisäisiin haasteisiin lukeutuu myös heikko tietomallien ylläpito ja dokumentoinnin heikko laatu (Hvam et al., 2019). Yrityksen sisäiset haasteet ovat siis kaikki liitettävissä johtamiseen. Johdon on aluksi annettava riittävät resurssit konfiguraattorin toteuttamiselle ja ylläpidolle sekä sen jälkeen johdettava ja opastettava käyttöönotto siten, että konfiguraattorille saadaan haluttu käyttöaste.

Yrityksen ja markkinan välisessä suhteessa saattaa esiintyä ongelmia, jotka vaikuttavat konfiguraattoriprojektin epäonnistumiseen. Esimerkiksi heikko viestintä kohderyhmän kanssa aiheuttaa huonon ymmärryksen mahdollisista asiakkaista ja sen seurauksena kohderyhmän määrittelemisen voi epäonnistua ja tuotteelle ei olekaan markkinoita. Toisaalta mahdollinen asiakasryhmä voi jäädä täysin tietämättömäksi tuotteen olemassaolosta, jos yrityksellä on heikko markkinointibudjetti tai huono kyky markkinoida tuotetta. (Blazek & Pilsl, 2017) Tätä osa-aluetta voi siis parantaa kehittämällä ymmärrystä omasta asiakaskunnasta sekä hankkimalla riittävät työkalut markkinan tutkimiseen.

3.2 Konfiguraattorin tuoma lisäarvo

Trentin et al. (2012) ovat tutkineet konfiguraattorien käyttöä 176 tuotantolaitoksen joukossa. Tuotantolaitosten toiminta sijoittuu kolmelle eri toimialalle kuudessa eri maassa. Heidän tutkimustuloksensa osoittavat, että tuotekonfiguraattoreiden käytöllä on positiivisia vaikutuksia tuotteen laatuun, mutta asiakaskunnan tarpeiden tunnistaminen vaikuttaa samaan aikaan tuotteen onnistumiseen negatiivisesti. Näiden laatuvaikutusten lisäksi Hvam et al. (2019) listaavat varianttien luomiseen käytetyn ajan vähenemisen,

rutiinitöiden automatisoitumisen, pienemmän läpimenoajan, paremman toimitusvarmuuden ja tiedon saatavuuden.

Laadun käsitteeseen on tarjolla useita eri selityksiä (Shewfelt, 1999; Trentin et al., 2012). Trentin et al. (2012) tarkastelevat laatua Garvinin kahdeksan laadun osa-alueen pohjalta. Nämä laadun peruselementit ovat suorituskky, ominaisuudet, luotettavuus, vaatimustenmukaisuus, kestävyys, huollettavuus, estetiikka ja havaittu laatu.

Ensimmäisenä tuotekonfiguraattori poistaa riskin siltä, että myyjä unohtaa inhimillisen virheen nimissä kysyä joitain tuotetta koskevia vaatimuksia tai tuoteominaisuuksia, eli tuote vastaa paremmin asiakkaan vaatimuksiin riippumatta myyntityötä tekevän henkilön tottumuksista myyntiprosessissa. Kestävyys ja luotettavuus paranevat samalla, kun asiakkaalle ei tarjotakaan myyjän vakioratkaisua, vaan tuote soveltuu käyttötarkoituksen mukaisiin vaatimuksiin ja rasitukseen. (Trentin et al., 2012) Samalla konfiguraattori myös helpottaa uusien myyjien perehdytystä, kun konfiguraattori määrittelee polun, jonka myyjä kulkee asiakkaan kanssa tuotteen ominaisuuksia valitessa.

Toinen laatua parantava vaikutus on tuotteen kokoonpanoon liittyvien virheiden vähentäminen. Trentin et al. (2012) havaitsivat, että 20% tuoreista ihmisen muokkaamista tuotevarianttien osaluetteloista sisälsi virheitä, jotka vaikuttivat kokoonpanon toimimattomuuteen ja sen seurauksena heikompaan toimitusaikaan, kestävyteen ja sen seurauksena heikompaan laatuvaikutukseen. Tuotekonfiguraattorilla tällaisten inhimillisten virheiden vaikutus saadaan minimoitua, mikä parantaa asiakkaan kokemaa ja suorituskyyllistä laatua.

Kolmantena laatua parantavana vaikutuksena tuotekonfiguraattorit vähentävät tarvetta luoda uniikkeja ratkaisuja asiakkaille. Tämä johtuu siitä, että tuotetietojen tehokas generointi asiakkaan asettamia ominaisuuksia vastaavaksi mahdollistaa yrityksen käyttämään samoja vakioituja ratkaisuja aina tietyissä asiakastarpeissa. Mikäli tuotetietomassaa ei pystytä tehokkaasti generoimaan, joudutaan luomaan aina uusia uniikkeja ratkaisuja ja samalle asiakkaalle saattaa päätyä samoilla kriteereillä tehdyistä tuotteista eri aikoina erilaisia lopputuloksia, mikä vaikuttaa koettuun laatuun negatiivisesti. (Trentin et al., 2012) Kun prosessiin saadaan toisteisuutta, myös kehitystyössä voidaan keskittyä yhä pienempiin yksityiskohtiin, millä saavutetaan parempaa kestävyttä tai yksinkertaisempia tuotantotapoja ja edullisempaa hintaa. Uniikkien ratkaisujen tekeminen on prototyyppien tekemistä, jolloin tuotekehitys ei kerkeä samoilla resursseilla keskittymään yhtä vahvasti tuotteen kehitykseen.

Neljäntenä tekijänä Trentin et al. (2012) nostavat tuotesuunnittelijoilta vapautuvan ajan, kun heidän ei tarvitse olla myyjien tukena antamassa teknisiä tietoja tuotteista myyntiprosessissa. Eräs tutkimuksessa mukana ollut sähkömuuntajia ja virtalähteitä valmistava yritys sai vähennettyä teknisen puolen henkilöstön työmäärää 70% tilauksen

aikana konfiguraattorin avulla. Tämän työpanoksen he voivat kehittää osa-alueita, joihin ei aiemmin ole riittänyt aikaa.

Viidentenä tekijänä Trentin et al. (2012) toteavat prosesseihin ja tuotteisiin liittyvien parannusten nopeamman käyttöönoton, kun ne sisällytetään osaksi tuotekonfiguraattorin ohjaamaa kanavaa. Pienten parannuksien tuominen käytäntöön vaatii muussa tapauksessa jotain toista erittäin hyvää informointikanavaa, jottei se unohdu myyjiltä myyntitilanteessa. Konfiguraattorin kaltainen kokonaisvaltainen kanava myyjille tuntuu kuitenkin selkeästi paremmalta vaihtoehdolta tiedon saavutettavuuden kannalta kuin useat erilliset viestit hyvässäkin kanavassa.

Kaikkien näiden positiivisten vaikutusten teho riippuu kuitenkin yrityksen kyvystä selvittää ja tulkita markkinoiden tarpeita. Mitä haastavampaa markkinoiden tarpeiden tulkinta on, sitä vähemmän konfiguraattorilla saadaan hyötyjä. Tämä voi johtua puutteellisesta markkinatiedosta, potentiaalisen asiakasryhmän määrittämisen haasteista, asiakkaiden ymmärtämättömyydestä omista tarpeistaan tai asiakkaiden kyvyttömyydestä ilmaista omia tarpeitaan. (Trentin et al., 2012) Tästä esimerkkinä voidaan pitää edullisia autoja kaikelle kansalle tuottanutta Henry Fordia, jonka asiakaskunta tuskin olisi osannut toivoa autoa. He olisivat luultavimmin halunneet nopeammat hevoscärryt. Asiakkailla ei siis välttämättä ole ymmärrystä kehittyvien tekniikoiden tuomista mahdollisuuksista ja siksi liian suoraviivainen asiakkaiden kuunteleminen voi myös viedä tuotekehityksen väärään suuntaan.

Tutkimuksensa lopputuloksena Trentin et al. (2012) saivat tukea molemmille hypoteeseilleen. Ensimmäinen hypoteesi oli: ”Tuotekonfiguraattorin käytöllä on positiivinen vaikutus tuotteen laatuun” ja toinen: ”Kun markkinoiden tarpeiden määrittämisen vaikeus kasvaa, tuotekonfiguraattorin käytön positiivinen vaikutus tuotteen laatuun heikkenee”. He kuitenkin toteavat laadunvalvonnan ja laatumääritelmän olevan vielä heikosti määriteltyjä massamuokkauksen näkökulmasta.

4. SOVELTAVA OSUUS

4.1 Soveltavan työn rajaaminen

Soveltavassa osuudessa luon konfiguraattorin hydraulisylinterituoteperheelle. Konfiguraattorin on tarkoitus antaa 3D-malli käyttäjän syöttämiä lähtötietoja vastaan. Konfiguraattorin käyttäjä syöttää järjestelmään käytettävän hydraulijärjestelmän vallitsevan öljynpaineen sekä halutun työntövoiman, joiden perusteella ohjelma laskee tarvittavan sylinterin sisähalkaisijan. Käyttäjä syöttää järjestelmään myös työpituuden, eli sylinterivarren uloimman ja sisimmän asennon välisen etäisyyden. Lisäksi voidaan valita kolmesta eri kiinnitysvaihtoehdosta käyttötarkoitukseen sopivin ja lisätä tarvittaessa kiinnitysnivel sylinterivarren päähän.

Soveltavan osuuden sylinterikonfiguraattori on yksinkertainen tuotekonfiguraattori, jota ohjataan Excelin avulla. Toteutan Excelin Visual Basic for Applications -ohjelman avulla käyttäjälomakkeen, johon käyttäjä syöttää lähtöarvot. Tämän jälkeen Excel laskee olennaiset mitat ja Siemens NX 12 ohjelma lukee arvot luodulle hydraulisylinterin kokoonpanomallille, minkä seurauksena käyttäjä saa 3D-mallin haluamastaan sylinteristä.

4.2 Konfiguraattorin luominen

Aloitin projektin luomalla mallinnusohjelmassa sylinterin, jossa kaikkien osien mitat pohjautuvat sylinteriputken halkaisijaan, jonka Excel syöttää mallinnusohjelmalle. Osien välisiä parametreja saadaan välitettyä toisille osille Siemens NX 12 -ohjelmassa Tools-valikon Expressions-välilehdellä olevasta Create Multiple Interpart Expressions -työkalun avulla, jolloin aktiivisena olevaan malliin saadaan vertailumittoja haetuista osista. Tätä työkalua käyttäen loin mallin siten, että kaikki osat skaalautuvat sylinteriputken sisähalkaisijan mukaan yhteensopiviksi. Loin mallin siten, että pienin mahdollinen sylinterin sisähalkaisija on 25mm. Pienempien arvojen syöttämisen estämiseksi asetan Excel soluun ehtolauseen, joka syöttää tätä pienempiin tarvittaviin voimiin kuitenkin 25mm sisähalkaisijalla varustetun sylinterin.

Hydraulisylinteri on mallina yksinkertainen, mutta silti mallinnettavia osia ja osien välisiä riippuvuuksia on sen verran paljon, että päätin luoda selkeyden vuoksi yksinkertaisen tuotetiedon hallinnan. Tuotetiedon hallinnan toteutin osaluettelona, josta

ilmenee jokaisen osan mitat ja niiden riippuvuus muiden osien mitoista, minkä voi nähdä kuvasta 3.

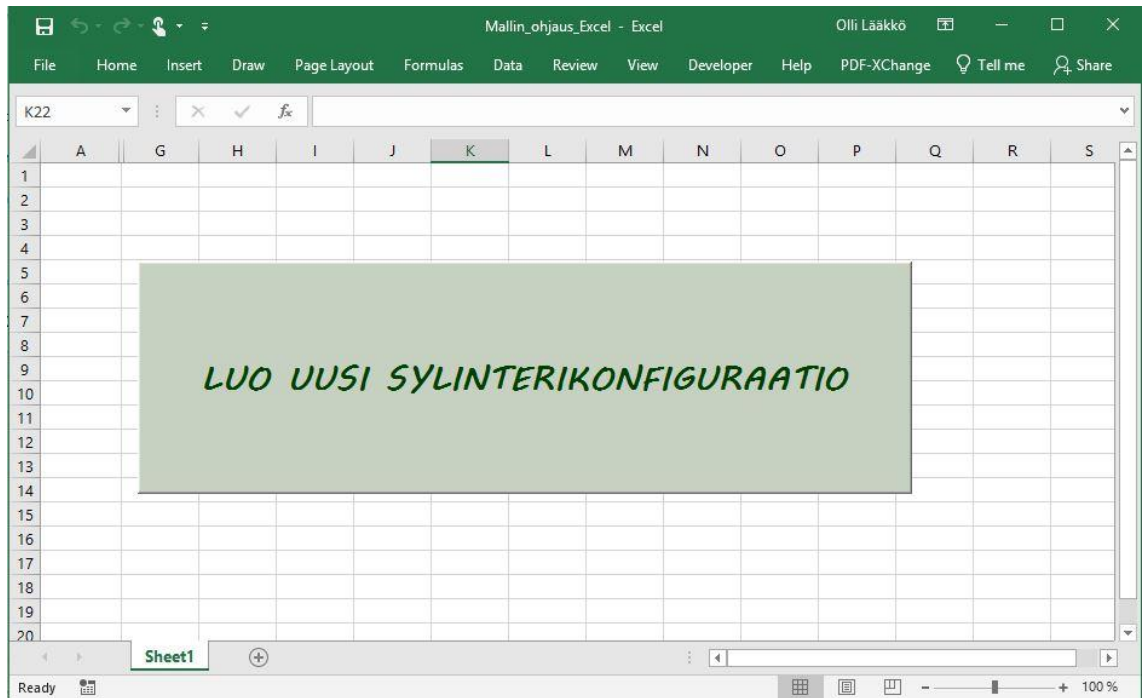
	A	B	C	D
1	Hydraylisylinterin BOM	Vapaa muuttuja	Lukittu muuttuja	Riippuvuussuhde
2		X		
3	Työpituus	X		
4	järjestelmän paine	X		
5	haluttu voima	X		
6	SYLINTERIPUTKI			
7	pituus		X	Työpituus + männän pituus
8	sisähalkaisija		X	$\sqrt{4 \cdot \text{haluttu voima} / \text{järjestelmän paine} / \pi}$
9	ulkohalkaisija		X	sylinteriputken sisähalkaisija + (6mm, jos SP5<50mm; 8mm, jos 50mm<SP5<150mm; 12mm, jos 150<SP5
10	MÄNNÄNVARSI			
11	pituus		X	sylinteriputken pituus + männän pituus + lukkomutteriosan pituus + varren ulkopuolel pituus
12	halkaisija		X	16, jos sylinteriputken sisähalkaisija <32; 25, jos 32<SP5<50; 35, jos 50<SP5<100; 50, jos 100<SP5<200; 100, jos 200<SP5
13	männän halkaisija		X	Mäntä, sisäreiän halkaisija
14	männän pituus		X	Mäntä, pituus
15	lukkomutteriosan halkaisija		X	Lukkomutteri, sisähalkaisija
16	lukkomutterin pituus		X	25mm
17	lukkosokan reijän etäisyys akselin päästä		X	3, jos lukkomutterin sisähalkaisija < 12; muulloin 4
18	varren ulkopuolel halkaisija		X	Sama kuin lukkosokan reijän halkaisija
19	varren ulkopuolel pituus		X	Männänvarren halkaisija - 5mm, jos männänvarren halkaisija < 40mm; muuten mhalkaisija-15mm
20	MÄNTÄ			
21	pituus		X	30mm, jos sylinteriputken sisähalkaisija < 40mm; 50mm, jos 40<SP5<120mm; 80mm, jos 120<SP5
22	halkaisija		X	Sama kuin sylinteriputken sisähalkaisija
23	sisäreiän halkaisija		X	10mm, jos sylinteriputken sisähalkaisija < 32mm; 16mm, jos 32<SP5<50mm; 22mm, jos 50<SP5<150mm; muuten 50mm
24	reunaolakeen pituus		X	9mm
25	Reunaolakeen halkaisija alapää		X	sisäreiän halkaisija + (6mm, jos sylinteriputken sisähalkaisija < 40; 10mm, jos 40<SP5<150mm; 25mm, jos 150<SP5
26	Reunaolakeen halkaisija yläpää		X	männänvarren halkaisija + 10mm, jos männänvarren halkaisija suurempi kuin reunaolakeen halkaisija alapää; muuten sama kuin reunaolaki
27	tiivisteolakeen leveys ja korkeus		X	3mm, jos sylinteriputken sisähalkaisija < 40mm; 5mm, jos 40<SP5<70mm; muuten 10mm
28	MÄNNÄN TIIVISTE			
29	leveys ja poikkileikkauksen korkeus		X	sama kuin tiivistelakeen leveys ja korkeus
30	ulkohalkaisija		X	Sama kuin sylinteriputken sisähalkaisija
31	MÄNNÄN KIINNITYSPRIKKA			
32	ulkohalkaisija		X	Sama kuin Männän reunaolakeen halkaisija alapää
33	sisäreiän halkaisija		X	Sama kuin männänvarren lukkomutteriosan halkaisija
34	paksuus		X	3mm

Kuva 3 Tuotetiedon hallinta Excelissä

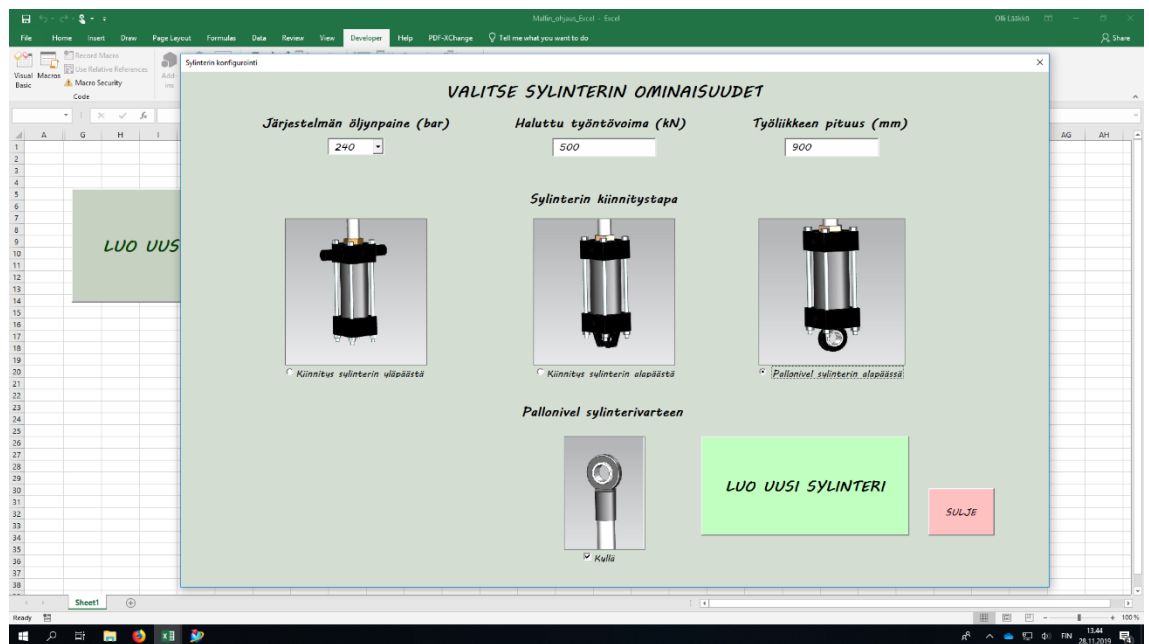
Excelin ja mallinnusohjelman välinen yhteys on luotu mallinnusohjelmaan. Toteutin omassa työssäni Export to Spreadsheet -toimintoa, jolla sain Expressions-välilehdelle keräämäni olennaiset mitat siirrettyä Excel-tiedostoon, joka on valmiiksi yhteydessä 3D-mallin kokoonpanoon. Kokoonpanon Expressions-välilehdellä oleviin mittoihin sain Excel-taulukon arvot ug_cell_read -komennolla. Nyt minulla on malli, jonka arvoja voin päivittää Excel-tiedoston kautta. Seuraavaksi luon Exceliin käyttäjäystävällisemmän rajapinnan VBA-ohjelmoinnin avulla.

Lomake kysyy käyttäjältä vaaditut alkuarvot. Lomakkeen luominen on VBA:ssa toteutettu graafisella käyttöliittymällä, mikä tekee ulkoasun järjestämisestä helppoa. Valikosta valitaan halutut kuvakkeet, syötekentät, valintapainikkeet ja napit. (Merensalmi, 2007 s.394-400)

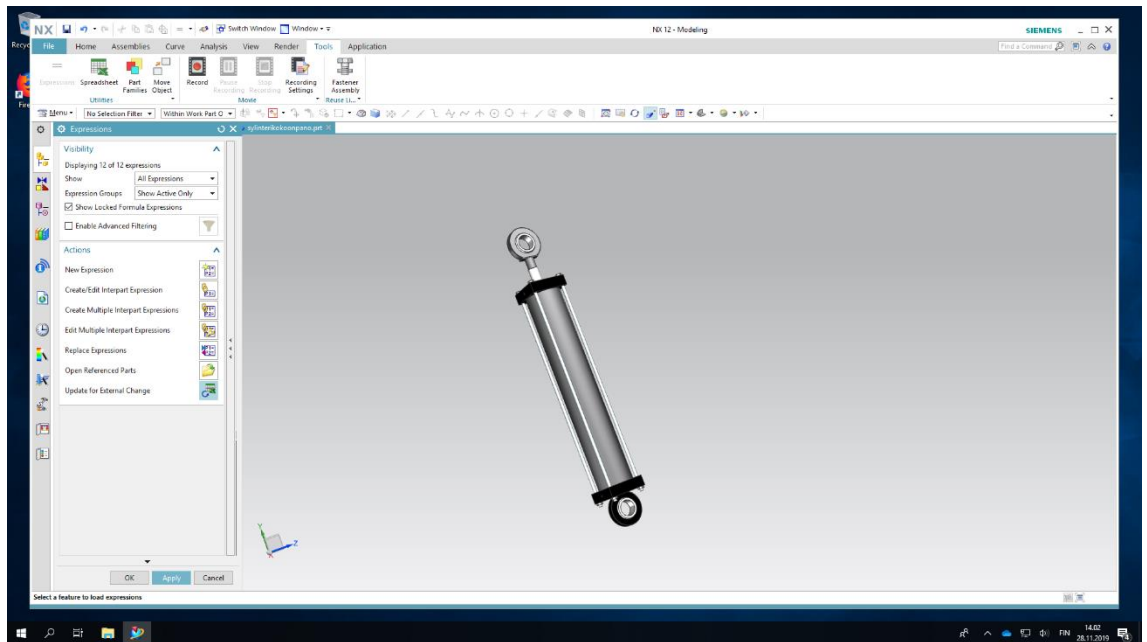
Ohjelmoin uuden konfiguraation tietoja kysyvän lomakkeen aukeamaan Excelissä olevasta napista, joka näkyy kuvassa 4. Tietojen syöttämisen jälkeen käyttäjä painaa "luo uusi sylinteri" -nappia, jolloin ohjelma vie annetut tiedot Exceliin ja laskee niiden pohjalta sisähalkaisijan sekä tallentaa Excel-tiedoston. Kuvassa 5 näkyy tarkemmin konfiguraattorin käyttöliittymä. Tämän jälkeen luotu konfiguraatio voidaan avata mallinnusohjelmassa Tools -> Expressions -> Update for External Change, jolloin uuden konfiguraation tiedot päivittyvät malliin (kuva 6).



Kuva 4 Näkymä Excelissä



Kuva 5 Konfiguraattorin käyttöliittymä



Kuva 6 Konfiguraation päivitys Siemens NX12 ohjelmassa

Tällainen konfiguraattori voisi olla käytössä sylintereiden myyntikonfiguraattorina tai suurempien kokonaisuuksien kasaamisen tapauksessa tuotekonfiguraattorina. Ensimmäisessä tapauksessa konfiguraattori helpottaa myyjän tai ostajan valintaprosessia ja vähentää tarvetta mallinnuksen tai teknisten tietojen syvemmälle ymmärtämiselle, kun tavanomaisilla järjestelmän suureilla saadaan luotua haluttu sylinterikonfiguraatio. Tuotekonfiguraattorina käytettäessä konfiguraattori vähentää suunnittelijan tarvetta suunnitella järjestelmässä usein esiintyviä sylintereitä. Tämä nopeuttaa suurten kokonaisuuksien suunnittelua ja antaa suunnittelijalle mahdollisuuden kohdistaa suunnitteluresurssit tärkeämpiin yksityiskohtiin.

5. YHTEENVETO

Mitä vaaditaan onnistuneeseen konfiguraattoriprojektiin? Tämän työn kautta avautui ymmärrys toimivan konfiguraattorin perustana olevasta tuotetiedon hallinnasta ja nimikkeiden hallinnan tärkeydestä kokonaisuuden toimintaan. Vaikka yrityksellä ei ole konfiguraattoreita, on hyvä tuotetiedon hallinta tärkeä osa tehokasta suunnittelua ja tuotantoa. Toimivalla tuotetiedon hallinnalla vältetään päällekkäinen suunnittelu ja samoihin asiakastarpeisiin luodut toisistaan poikkeavat ratkaisut, jolloin yrityksen hallinnassa on vähemmän nimikkeitä ja tuotanto voi toistaa hyväksi koettuja tuotemalleja.

Onnistunut konfiguraattori vaatii oikean päättelymallin. Oikean päättelymallin valinta riippuu konfiguraattoria käyttävästä kohderyhmästä ja tuotteen monimutkaisuudesta. Malliin perustuva päättely antaa sääntöihin perustuvaa helpomman ja monipuolisemman hallinnan, kun taas tapauksiin perustuva on helppo valinta, mikäli omistetaan suuri datamäärä jo tapahtuneita variointeja ja variointeissa ei olla luomassa uutta.

Itse projektin kannalta on oleellista, että yritys antaa riittävät resurssit konfiguraattorin suorittamiseen ja projektilla on vastuuhenkilö, joka vie määrätietoisesti projektia eteenpäin. Riittävän hyvä markkinatuntemus on tärkeää, jotta konfiguraattorin kehitys suuntautuu alusta alkaen oikeaan kohderyhmään. Johtamisen merkitys on suuri projektin etenemisen kannalta. Liian hitaasti edetessään projekti sakkaa ja yrityksen sisäinen luottamus projektista saatuihin hyötyihin voi loppua.

Mitä onnistuneella konfiguraattoriprojektilla voidaan saavuttaa? Parhaimmillaan konfiguraattorit voivat parantaa tuotteiden laatua usealla eri mittarilla. Asiakkaat saavat paremmin omiin tarpeisiinsa sopivat tuotteet, jolloin he kokevat tuotteen laadukkaampana. Tuotannossa tehdään enemmän toistoja samoista tuotteista, jolloin kokemus ja ymmärrys parantaa tuotteen kestävyyttä sekä tuotanto nopeutuu. Tuotekehittäjien aikaa vapautuu myyntityön tukemisesta tuotteiden jatkokehitykseen, jolloin voidaan keskittyä täsmällisempiin yksityiskohtiin tuotteissa.

Soveltavan osuuden myötä ymmärsin, kuinka pienessäkin tuotekonfiguraattorissa osia ja niiden välisiä yhteyksiä on niin paljon, että kokonaisuuden hahmottaminen ilman tuotetiedon hallintaa on vaikeaa. Soveltava osa paransi ymmärrystä myös Siemens NX:n ja Excelin välisestä yhteydestä sekä Excel VBA:n käytöstä. Tuotetiedon hallinnan ja konfiguraattorin ylläpito pienten muutoksien päivittämisessä vaatii työtä. Yrityksessä on siis tiedostettava konfiguraattorilla saavutettavissa olevat hyödyt ja vaaditut resurssit ja riskit, jotta projektiin lähtiessä on realistinen käsitys projektin läpiviennin vaikutuksista.

Mikäli yrityksellä on mahdollista luoda tuotteisiinsa vain muutama standardoitu variaatio, on se järkevämpää, kuin konfiguraattorin luominen kyseisten varianttien luomiseksi, sillä yrityksen hallintaa vaativa tietomäärä on standardointitapauksessa huomattavasti pienempi. Tällöin myös tuotteeseen tulevat muutokset vaikuttavat pienempään tuotejoukkoon, mikä yksinkertaistaa tuotannon ja myynnin lisäksi varaosajakelua ja huoltoa.

LÄHTEET

Blazek, P., & Pilsl, K. (2017). LEARNINGS FROM SETTING UP PRODUCT CONFIGURATOR PROJECTS - ProQuest. *Annals of the Faculty of Engineering Hunedoara*, 15(1), 25-28.

Saatavissa [www-muodossa:https://search-proquest-com.lib-proxy.tuni.fi/docview/1881067043?pq-origsite=summon](https://search-proquest-com.lib-proxy.tuni.fi/docview/1881067043?pq-origsite=summon)

Bonev, M., Hvam, L., Clarkson, J., & Maier, A. (2015). Formal computer-aided product family architecture design for mass customization. *Computers in Industry*, 74, 58-70. Saatavissa [www-muodossa:https://www.sciencedirect.com/science/article/pii/S0166361515300245](https://www.sciencedirect.com/science/article/pii/S0166361515300245)

Crnkovic, I., Asklund, U., & Dahlqvist, A. P. (2003). *Implementing and integrating product data management and software configuration management*. Norwood: Artech House INC.

Haug, A., Hvam, L., & Mortensen, N. H. (2012). Definition and evaluation of product configurator development strategies. *Computers in Industry*, 63(5), 471-481. doi:10.1016/j.com-pind.2012.02.001

Hvam, L., Kristjansdottir, K., Shafiee, S., Mortensen, N. H., & Herbert-Hansen, Z. N. L. (2019). The impact of applying product-modelling techniques in configurator projects. *International Journal of Production Research*, 57(14), 4435-4450. doi:10.1080/00207543.2018.1436783

Liu, T., & Xu, W. (2001). A review of web-based product data management systems. *Computers in Industry*, 44(3), 251-262. doi:10.1016/S0166-3615(01)00072-0

Luukkainen, M., & Laine, H. (2010). *Ohjelmistojen mallintaminen* (2nd ed.). Helsinki: Helsingin Yliopisto.

Martio, A. (2015). * tuotekonfigurointi ja tuotetiedon hallinta* (1st ed.). Kurikka: Amartekno Oy.

Merensalmi, J. (2007). In Jutila T., Kolari L. and Tolonen E. (Eds.), *Excel VBA yrityskäytössä* (1st ed.). Porvoo: WS Bookwell.

Peltonen, H., Martio, A., & Sulonen, R. (2002). *Tuotetiedon hallinta* (1st ed.). Helsinki: Edita Prima Oy.

Selladurai, R. S. (2004). Mass customization in operations management: Oxymoron or reality? *Omega*, 32(4), 295-300. doi:10.1016/j.omega.2003.11.007

Shewfelt, R. L. (1999). What is quality? *Postharvest Biology and Technology*, 15(3), 197-200. doi:10.1016/S0925-5214(98)00084-2

Trentin, A., Perin, E., & Forza, C. (2012). Product configurator impact on product quality. *International Journal of Production Economics*, 135(2), 850-859. doi:10.1016/j.ijpe.2011.10.023

Wang, Y., Mo, D. Y., & Tseng, M. M. (2015). ScienceDirect. *Clinical Microbiology Newsletter*, 37(4), 33. doi:10.1016/j.clinmicnews.2015.01.008